Algorithmique – Python - Mathématiques

ALGORITHMIE

1 – Bases

1.1 Simple échange

v1 : entier

v2 : entier

temp : entier

v1 <- 1

v2 <- 2

temp <- 0

temp <- v1

v1 <- v2

v2 <- temp

1.2 Le carré

val1 : entier

val1 <- 0

LIRE(val1)

AFFICHER("Le carré de", val1, " est " val1\*val1)

1.3 La condition

val1 : entier

val1 <- 0

LIRE(val1)

SI val1 > 0 ALORS

AFFICHER("Le nombre entré est positif")

SINON

AFFICHER("Le nombre entré est négatif")

FIN\_SI

1.4 Le produit

val1 : entier

val2 : entier

val1 <- 0

val2 <- 0

LIRE(val1)

LIRE(val2)

SI val1 \* val2 > 0 ALORS

AFFICHER("Le produit sera positif")

SINON

AFFICHER("Le produit sera négatif")

FIN\_SI

1.5 Calcul

val1 : entier

i : entier

LIRE(val1)

POUR i <- 0 ALLANT A 10

AFFICHER(val1+i)

FIN\_POUR

1.6 Additivité

val1 : entier

somme : entier

val1 <- 0

somme <-0

LIRE(val1)

POUR i <- 1 ALLANT A val1

somme <- i

AFFICHER(somme)

FIN\_POUR

2 Les tableaux

2.1 La somme

tab : tableau de taille N

somme : entier

somme <- 0

POUR i <- 0 ALLANT A N-1

somme <- somme + tab[i]

FIN\_POUR

2.2 La somme : retour

tab : tableau de taille N

tab2 : tableau de taille N

tab3 : tableau de taille N

POUR i <- 0 ALLANT A N-1

tab3[i] <- tab[i] + tab2[i]

FIN\_POUR

RAPPEL MATHEMATIQUES

Produit de Matrice :

A = B = C =

C = A.B

Prouver que A.B différent de B.A

On dit que D est la matrice résultante du produit B.A

D = B.A = \* =

C étant différent de D par conséquent B.A différent de A.B

SOFT SKILLS ADVICE

* LinkedIn a jour
* Biographie
* Cartographie par typologie ou par secteur

MATHS

Fonction tempArret(alpha : float) -> int :

n = 0

k = 1

tant que somme < alpha faire

somme += 1/k

k += 1

fin\_tant\_que

si somme > alpha

n = k

fin\_si

PRODUIT MATRICIEL :

k <- 0

somme <- 0

i <- 0

j <- 0

matA <- [

[0,1,3],

[0,1,2],

[0,1,2]

] de taille N

matB <- [

[0,1,3],

[0,1,2],

[0,1,2]

] de taille N

matC <- [[][][]] de taille N

POUR i ALLANT A N FAIRE

POUR j ALLANT A N FAIRE

POUR k ALLANT A N FAIRE

somme += matA[i][k]\* matB[k][j]

FIN\_POUR

matC[i][j] = somme

FIN\_POUR

FIN\_POUR

MatA[0][0]\*MatB[0][0]+MatA[0][1]\*MatB[1][0]+MatA[0][2]\*MatB[2][0] = MatC[0][0]

MatA[0][0]\*MatB[0][1]+MatA[0][1]\*MatB[1][1]+MatA[0][2]\*MatB[1][2] = MatC[0][1]

CALCUL INVERSE MATRICIEL

A =

detA = 1/(1\*4)-(3\*2) = 1/-2

A-1 = 1/-2\*

A-1 =

Soit M =  ; Calculer (M – lambda\*Id2) et trouver u1 et u2 tel que U = ou (M-lambda\*Id2) \* =

- =

= det()

= -lambda1(3-lambda2) – (-2\*1)

= lambda² - 3lambda +2

Calcul delta :

Delta = b² - 4ac

= 3² -4\*1\*2

= 9-8

= 1

Si delta > 0 alors

-b + racine de lambda / 2a et -b – racine de lambda / 2a

Lambda1 = 1

Lambda2 = 2

PROBABILITE

P(A) = (nb eventualité def – A) / (nb total d’eventualite Ω)

= | A | / | Ω |

P(A OU B)

P(A ET B)

Si A et B indépendant alors P(A\*B) = P(A) \* P(B)